Ciencia e Ingeniería
Revista Interdisciplinar de Estudios en Ciencias Básicas e Ingenierías.
Año 2015, Julio-Diciembre, Vol. (2) N° (2) ISSN 2389-9484.
Universidad de La Guajira, Facultades de Ciencias Básicas y Aplicadas e Ingeniería.
La Guajira-Colombia.
Revista en Línea http://revistas.uniguajira.edu.co/index.php/cei



SISTEMA DE CONTROL POR GPRS A TRAVES DEL MODULO SM5100B-D

GPRS CONTROL SYSTEM THROUGH THE MODULE SM5100B-D

Salomón Manosalva[1]y Sergio Velásquez[*]

Centro de Investigación de las Redes Neuronales y la Robótica CIRNAR-UNEXPO
Ciudad Guayana, Bolívar 8050 Venezuela.
shlomojoseph@gmail.com

Recibido: Julio 21 de 2015 Aceptado: Noviembre 11 de 2015

RESUMEN

En la actualidad vivimos en un mundo globalizado gracias al avance de la tecnología en las últimas décadas, por lo que se ha vuelto una necesidad el tener cerca el control de todo lo que operamos o manipulamos en nuestras vidas. El objetivo principal de esta investigación es diseñar un sistema de control que mediante el manejo de paquete de datos (GPRS) y usando el módulo shield SM5100B-D, un operador o usuario a través de internet pueda hacer uso de una página web para monitorear y manipular de manera interactiva la velocidad de un motor AC. Por lo cual este proyecto se considera una investigación proyectiva cuyo fin es desarrollar un sistema funcional que permita solventar una necesidad

Palabras Clave: Módulo GPRS, Arduino, monitoreo, comandos AT.

ABSTRACT

Today we live in a globalized world thanks to advances in technology in recent decades, so has become a necessity to have close control of everything that we operate or manipulate in our lives. The main objective of this research is to design a control system by handling packet data (GPRS) and using the module shield SM5100B-D, an operator or user through internet can make use of a web page to monitor and interactively manipulate a speed AC motor. So this project is considered a projective research aimed at developing a functional system that allows solve a need.

Keywords: GPRS Module, Arduino, monitoring, AT commands.

---

1. INTRODUCCIÓN

Hoy en día los sistemas de supervisión y control en recintos domésticos, industriales, comerciales y donde se practica ingeniería se han vuelto una necesidad básica. Con el avance de la tecnología existen diversas formas de comunicación entre los dispositivos electrónicos y los usuarios, lo cual genera una gran comodidad y accesibilidad al usuario por medio del control a distancia de los dispositivos, ya que se podría monitorear y controlar mediante algún equipo electrónico el funcionamiento de un motor sin la necesidad de programarlo con anterioridad.

El método de diseño propuesto en este artículo radica en la posibilidad de la obtención de un esquema de control para que la dinámica del esclavo, así como la dinámica del estado del maestro y del esclavo, tengan el comportamiento deseado algo diferente a las perspectivas de control expuestas en función de la base de Fujimoto y Tanaka, los cuales realizan la convergencia en sistemas bilaterales a través de compensación de parada y ganancia (Sano *et all.* 1998), y de las características dinámicas del maestro, del esclavo y del entorno, debido al uso de una plataforma que para aquel entonces era inexistente

Una de las últimas técnicas de automatización sencilla más usada es la domótica, que se puede definir como la integración de la tecnología en el diseño inteligente de un recinto cerrado, este concepto se refiere a la automatización y control (encendido/apagado, apertura / cierre y regulación) de aparatos y sistemas de instalaciones eléctricas y electrotécnicos (iluminación, climatización, persianas y toldos, puertas y ventanas motorizadas, el riego, etc.) de forma centralizada y/o remota.

Esta técnica reciente busca implementar diferentes dispositivos con los cuales se logre realizar la supervisión y control del sistema. Motivo por el cual se desarrollan aplicaciones que favorecen al proceso desde dispositivos como, computadores, teléfonos inteligentes y tabletas.

En la actualidad, existen diversas formas de comunicación mediante la ayuda de la tecnología, en donde los operadores en telefonía móvil han desarrollado en sus servicios diferentes formas de comunicación entre los usuarios por medio de equipos móviles. Una de estas ideas fue la GPRS, que es una tecnología que comparte el rango de frecuencias de la red GSM utilizando una transmisión de datos por medio de paquetes, a la que se denomina hoy en día como “paquete de datos”.

En tal sentido este trabajo está enmarcado dentro de los proyectos que se están llevando a cabo en CIRNAR, ya que se tiene la necesidad de un sistema que presente factibilidad y libertad a la hora de controlar a distancia equipos electrónicos o eléctricos como lo es un motor AC de 110V y 500 W. De tal manera que se busca implementar diferentes dispositivos como el módulo GSM/GPRS, con los cuales se logre realizar la supervisión y control del sistema.

II. DESARROLLO

A. Protocolo HTTP

El protocolo HTTP es el que da vida a Internet, y gracias al cual, los clientes y servidores se pueden comunicar. Sus siglas significan Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999 con la publicación de una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse.

Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador web o un spider) se lo conoce como “user agent” (agente del usuario).

A la información transmitida se la llama recurso y se la identifica mediante un localizador uniforme de recursos (URL). El resultado de la ejecución de un programa.

B. Sistema GPRS

GPRS es una nueva tecnología que comparte el rango de frecuencias de la red GSM utilizando una transmisión de datos por medio de ‘paquetes’. La conmutación de paquetes es un procedimiento más adecuado para transmitir datos, hasta ahora los datos se habían transmitido mediante conmutación de circuitos, procedimiento más adecuado para la transmisión de voz.

C. Propuesta de diseño

Se desea diseñar un sistema de control monitoreable de un motor AC de 110V/500W (Figura 1), para lo cual se usara un Arduino UNO el cual posee una lógica programada que controla dicho motor y la comunicación con el shield SM5100B-D, este shield se comunica a la red mediante GPRS con una base datos MySQL que recibirá la data enviada por el shield, de igual manera esta posee la interfaz web con la cual el usuario podrá interactuar, de tal forma que este pueda enviar y recibir información.



Figura 1: Propuesta de diseño para el sistema de control.

D. Uso del Módulo shield SM5100B-D

Para el correcto funcionamiento del módulo GSM/GPRS al inicializarlo el fabricante ofrece una serie de códigos o señales que indican en qué estado se encuentra el mismo para luego poder realizar algún tipo de comunicación. Como son los siguientes:

Tabla 1 Código SIND y significado

+SIND=1
+ SIND: 10, „SM“, 1, „FD“, 1, „LD“, 1, „MC“, 1, „RC“, 1, „ME“, 1
+SIND=11
+SIND=3
+SIND=4

+SIND	Significado
0	Tarjeta SIM eliminada
1	Tarjeta SIM insertada
2	Anillo de melodía
3	AT módulo parcialmente listo
4	AT módulo está totalmente listo
5	Identificación de llamadas liberadas
6	Llamada liberada cuyo ID = <idx>
7	El servicio de red está disponible para una llamada de emergencia
8	La red se pierde
9	Audio ON
10	Muestre el estado de cada guía telefónica después que inicia la frase
11	Registrado en la red

La comunicación con el módulo comienza después de la primera línea que se muestra +SIND =1, la segunda línea +SIND = 10 nos dice si el módulo puede ver una tarjeta SIM, si la tarjeta se detecta todos los demás campos de la segunda línea son 1, sino se detecta los demás campos son 0. Después del +SIND 10 la respuesta del módulo se iniciara automáticamente tratando de conectarse a una red, por lo que se debe esperar una conexión de red antes de iniciar con el envío de comandos, estas respuestas son las 3era, 4ta y 5ta línea que se muestran, las cuales significan que el módulo está totalmente listo para comunicare y que ya se ha conectado a la red y puede comenzar la comunicación.

La respuesta +SIND del módulo celular indica el estado de los módulos. A continuación se muestra una tabla con los significados de cada respuesta que le pueden aparecer en el monitor serial, cuando se esté tratando de realizar la comunicación:

Tabla 2- Comando TCP-IP MÓDULO SM5100B-D

AT+SDATACONF	Configura los parámetros de la conexión TCP / UDP
AT+SDATASTART	Inicia la conexión TCP / UDP
AT+SDATATSEND	Enviar los datos UDP / TCP (longitud máxima transparente es 1460 bytes)
AT+SDATASEND	Envíe los datos TCP / UDP (formato HEX)

AT+SSTRSEND	Envíe los datos TCP / UDP (formato String)
AT+SDATASTATUS	Consultar el estado del socket y el TCP enviado / reconocido / bytes recibido
AT+SDATAREAD	Lee el dato recibido por UDP/TCP
AT+SDATATREAD	Leer los datos (transparente) TCP / UDP
AT+SDATARXMD	Ajusta el modo recibido (hex o formato string)
AT+TRT	Establece los tiempos máximos de reintento TCP

Las características de TCP / IP proporcionadas por los comandos AT para el SM5100B-D están destinados a proporcionar un subconjunto de las características que normalmente están disponibles en el nivel de toma cuando se utiliza una pila TCP / IP convencional, con algunas simplificaciones y personalizaciones en base a las características específicas de la SM5100B-D.

Los comandos permiten al escritor aplicación para crear sockets TCP, para controlar contextos PDP GPRS ya configurados, para transferir datos hacia y desde el módulo, y para interrogar a la información del estado de IP sobre el enlace activo. La interfaz de socket es proporcionado por una serie de comandos AT para la conexión TCP/IP que se describen en la Tabla 2.

La Data que se observara en la página web estará contenida en un servidor que posee una base de datos que almacenara las variables a supervisar (Temperatura, velocidad) enviadas por el Shield SM5100B-D a través de una petición HTTP 1.1, y los códigos PHP que generan la visual (botones de control, muestra de Temperatura y velocidad) que manipulara el usuario u operador del sistema.

Dependiendo de la empresa de telefonía que se use el APN varía:

- APN = int.movilnet.com.ve (MOVILNET)
- APN = internet.movistar.ve (MOVISTAR)
- APN = gprsweb.digitel.ve (DIGITEL)

Cada una de estos datos deben ser configurados en el código del Arduino con los comandos presentados anteriormente en este mismo capítulo. De igual forma la petición HTTP 1.1 que se realiza al servidor debe ser configurada, ya que esta se realiza mediante el método GET es definida en el Arduino UNO como:

GET/test/insertrow.php?tmp=”+String(temp)+”&vel=”+String(vel)+” HTTP/1.1

Donde String(temp) y String(vel) son los datos de temperatura y velocidad convertidos en una cadena para poder ser enviados mediante esta petición al servidor web.

E. Diseño del lado del servidor

Para poder realizar el sistema de control por paquete de datos e s necesario crear una página web ya que la información enviada y solicitada se realiza mediante peticiones HTTP 1.1, por lo que es necesario poseer un servidor de hosting con el que se pueda establecer una comunicación y de esta manera almacenar la data enviada por el módulo SM5100B-D. Por lo cual es necesario crear una serie de archivos PHP que permitan al usuario u operador ver la data recibida y enviar órdenes, para lograr el control del sistema (velocidad del motor AC).

Se crea una página de inicio para que solo el operador que posea un usuario y clave en el servidor pueda acceder, evitando que terceros intenten alterar la operatividad del sistema. Para esto se debe crear una Base de datos usando MySQL que contendrá el registro de usuario, botones y valores de velocidad y temperatura que serán recibidos desde el módulo SM5100B-D (figura 2).



Figura 2. Base de datos creada en MySQL usando phpMyAdmin.

F. Modelo del Sistema

El sistema maestro y esclavo, respectivamente, se representan en el espacio de estados de la siguiente forma.

$$\dot{x}_m(t) = A_m x_m(t) + B_m u_m(t)$$
$$y_m(t) = c_m x_m(t)$$
$$\dot{x}_s(t) = A_s x_s(t) + B_s u_s(t)$$
$$y_s(t) = c_s x_s(t)$$

donde:
• x_m y x_s son las variables de estado,
• y_m y y_s representan las salidas,
• A_m y A_s son las matrices de estado,
B_m y B_s son las matrices de entrada,
y C_m y C_s son las matrices de salida.

Las matrices que aparecen en el modelo son las siguientes:
G2: Define la influencia de la fuerza ejercida por el operador sobre el maestro en el esclavo.

Rm: Define la interacción esclavo – maestro.
Rs: Define la interacción maestro – esclavo.
Km: Matriz de realimentación de estado del maestro.
Ks: Matriz de realimentación de estado del esclavo.

Cabe resaltar que la topología de NCS que se muestra en Figura 1 en la que se consideran todas las posibles interacciones que pueden aparecer en el conjunto operador – maestro – esclavo – entorno, supone una idea innovadora en el modelado de sistemas de control NCS con retardos.

Análisis del modelo

Se va a realizar el análisis suponiendo que el maestro y el esclavo constan de un único grado de libertad, Figura 2. El modelo dinámico de un elemento de un único grado de libertad es el siguiente:

$$J\ddot{\theta}(t) + b\dot{\theta}(t) + mgl\sin\theta\,(t) = u(t)$$

donde $J = \frac{1}{3}ml$ es la inercia del elemento, b es el coeficiente de rozamiento viscoso, l es la longitud del elemento, g es la aceleración de la gravedad, m es la masa del elemento, $\theta(t)$ representa la posición (ángulo girado) en el instante t y $u(t)$ es el par de control aplicado en el instante t (Cho, 1999).

Figura 3. Modelo simplificado de un robot de 1 GDL.

Para efectuar el análisis, se va a considerar el modelo dinámico simplificado lineal:

$$J\ddot{\theta}(t) + b\dot{\theta}(t) = u(t)$$

Se va a expresar el modelo dinámico en el espacio de estados mediante las ecuaciones:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t)$$

donde el vector de estados es de la forma:

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

Para modelar el sistema en el espacio de estados se van a elegir variables de estado con sentido físico. En concreto se van a considerar como variables de estado la posición y la velocidad del elemento.

Sea $x_1(t) = \theta\ (t)\ y\ x_2\ (t) = \dot{x}_1(t)\ = \dot{\theta}(t)$, es decir, el estado x1 representa la posición del elemento, y el estado x2, la velocidad. A partir de la ecuación (6) se va a calcular la ecuación de estados (F.-L. Lian, J. Moyne, and D. Tilbury, 2007) (Ferrell, 1986) (C. E. García, R. Carelli, J. F. Postigo and B. Morales, 2000).

$$\dot{x}_1(t) = x_2(t) = \dot{\theta}(t)$$
$$\dot{x}_2(t) = \ddot{x}_1(t) = \ddot{\theta}(t)$$
$$= -\frac{b}{j}\dot{\theta}(t) + \frac{1}{J}u(t)$$
$$= -\frac{b}{J}x_2(t) + \frac{1}{J}u(t)$$
$$y(t) = x_1(t)$$

Por tanto, el modelado en variables de estado con sentido físico del sistema es:

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -\frac{b}{J}\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{J}\end{bmatrix}u(t)$$
$$y(t) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}$$

De esta forma, la representación en el espacio de estados del sistema maestro es (Coiffet, 1990):

$$\begin{bmatrix}\dot{x}_{m1}(t)\\ \dot{x}_{m2}(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -\frac{b_m}{J_m}\end{bmatrix}\begin{bmatrix}x_{m1}(t)\\ x_{m2}(t)\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{J_m}\end{bmatrix}u_m(t)$$
$$y_m(t) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_{m1}(t)\\ x_{m2}(t)\end{bmatrix}$$

La representación en el espacio de estados del sistema esclavo es:

$$\begin{bmatrix}\dot{x}_{s1}(t)\\ \dot{x}_{s2}(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -\frac{b_s}{J_s}\end{bmatrix}\begin{bmatrix}x_{s1}(t)\\ x_{s2}(t)\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{J_s}\end{bmatrix}u_s(t)$$
$$y_s(t) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_{s1}(t)\\ x_{s2}(t)\end{bmatrix}$$

De forma simplificada se representará el sistema maestro y esclavo, respectivamente, del siguiente modo:

$$\dot{x}_m(t) = A_m x_m(t) + B_m u_m(t)$$
$$y_m(t) = c_m x_m(t)$$
$$\dot{x}_s(t) = A_s x_s(t) + B_s u_s(t)$$
$$y_s(t) = c_s x_s(t)$$

donde:

$$A_m = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b_m}{J_m} \end{bmatrix} \quad B_m = \begin{bmatrix} 0 \\ \frac{1}{J_m} \end{bmatrix}$$

$$C_m = [1 \quad 0] \quad x_m(t) = \begin{bmatrix} x_{m1}(t) \\ x_{m2}(t) \end{bmatrix}$$

$$A_s = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b_s}{J_s} \end{bmatrix} \quad B_s = \begin{bmatrix} 0 \\ \frac{1}{J_s} \end{bmatrix}$$

$$C_s = [1 \quad 0] \quad x_s(t) = \begin{bmatrix} x_{s1}(t) \\ x_{s2}(t) \end{bmatrix}$$

La estructura del resto de matrices que aparecen en el modelo es la siguiente:

$$K_m = [k_{m1} \quad k_{m2}]$$
$$K_s = [k_{s1} \quad k_{s2}]$$
$$R_m = [r_{m1} \quad r_{m2}]$$
$$R_s = [r_{s1} \quad r_{s2}]$$
$$G_2 = g_2$$

A partir del modelo puede observarse que la señal de control del maestro, $u_m(t)$, y del esclavo, $u_s(t)$, vienen dadas por:

$$u_m(t) = K_m x_m(t) + R_m x_s(t-T) + F_m(t)$$
$$u_s(t) = K_s x_s(t) + R_s x_m(t-T) + G_2 F_m(t-T)$$

Si se sustituye la señal de control en las ecuaciones de estado (20) y (21) por las expresiones (27) y (28) se obtienen las siguientes ecuaciones de estado:

$$\dot{x}_m(t) = [A_m + B_m K_m] x_m(t) + B_m R_m x_s(t-T) + B_m F_m(t)$$
$$\dot{x}_s(t) = [A_m + B_m K_m] x_m(t) + B_m R_m x_s(t-T) + B_m G_2 F_m(t-T)$$

G. Control del Motor

La red eléctrica que tenemos disponible en la mayoría de los hogares en Venezuela es sinusoidal con 120 voltios, en el que la frecuencia de oscilación es de 60 Hz. Esto significa que tenemos un ciclo cada 16,66 ms. Para implementar el control del motor AC, se utilizó un TRIAC o triodo para corriente alterna es un componente equivalente electrónico a dos rectificadores controlados por silicio (SCR o tiristor) conectado en el terminal "puerta" antiparalela y el gatillo unidos entre sí. Este tipo de resultados de la vinculación en un interruptor electrónico bidireccional que puede conducir la corriente eléctrica en ambas direcciones.

Con el fin de activar el TRIAC en el momento adecuado, hay que sincronizar el disparo con la forma de onda de la red eléctrica, para ello se usa una vía de flujo de detector de cero voltios, en este trabajo este circuito está formado por las resistencias de 33KΩ, 10KΩ y el optoacoplador H11AA1 (figura 4). Cada semiciclo que pasa por cero voltios genera una interrupción (INT0) en el ATMEGA del Arduino, por lo que tiene una interrupción cada 8.33 ms. Para lo cual la programación establecida en el microcontrolador debe generar un impulso en la puerta del TRIAC en un rango de 150 microsegundos a ocho milisegundos después de la interrupción, por lo que se va a conducir la corriente eléctrica del medio ciclo actual, ya sea positivo o negativo. Cuanto más es el avance o porcentaje de potencia que se le asigne al disparar el TRIAC mayor será velocidad del motor (no hay que disparar el TRIAC exactamente después una interrupción, ya que la tensión será baja y no habrá todavía suficiente corriente para mantener la conducción del mismo, así que se deja un margen de 150 microsegundos).

El diseño del circuito de carga para disparar el TRIAC está formado por un optotriac (MOC3020), 220Ω y TRIAC NTE56016. Recordando que tanto el H11AA1 y el MOC3020 también tiene la función de aislar el circuito de corriente continua del circuito de corriente alterna. Para controlar el instante de disparo del TRIAC cuando el ATMEGA recibe una interrupción externa (INT0), se inicia TIMER1 con un valor en microsegundos que es inversamente proporcional a la última lectura válida al valor asignado por software, una vez transcurrido el periodo asignado se generará una interrupción que indica la orden de disparo del TRIAC. Al momento de realizarse el disparo se define la intensidad o porcentaje de potencia que tendrá el motor. La implementación de utilizar el TIMER1 como el conmutador o generador de disparos se realiza utilizando la librería TIMER1.

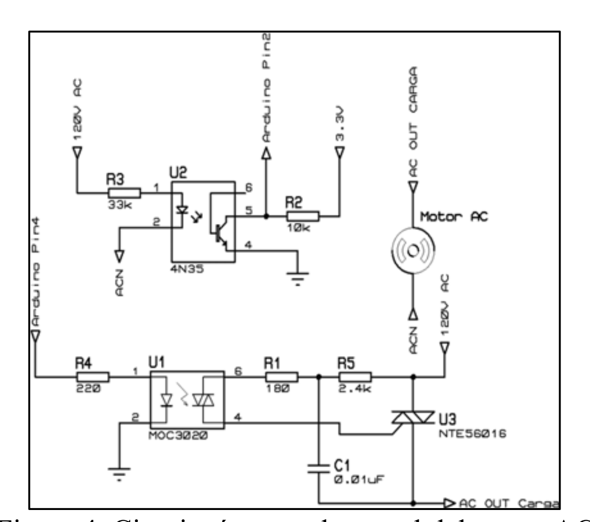


Figura 4. Circuitería para el control del motor AC

Nota: *El integrado H11AA1 es un optoacoplador con dos diodos emisores de luz en conexión antiparalelo, puede ser sustituido por un 4N35 con la excepción de que la detección de cruce por cero solo se hará en el semiciclo positivo de la señal AC entrante.*

Para proteger al motor de un futuro daño por recalentamiento se implementa una protección térmica con el sensor de temperatura LM35 (figura 5). Para cual se configura una lectura de este sensor cada un segundo con el TIMER2 del ATMEGA. El sensor está conectado al pin A0 del Arduino, para obtener el valor de la temperatura en grados centígrados (°C) se usa la siguiente formula:

temp = (5.0 * analogRead(PIN_SENSOR) * 100.0) / 1024

Esta fórmula se deduce de la relación del sensor con los grados. El sensor de temperatura LM35 según el datasheet responde a variaciones de 10 mV por cada grado centígrado. Si el sensor detecta 1 grado centígrado a la salida del sensor obtendríamos 10 mV. Ejemplo: 26,4ºC = 264 mV = 0.264 V.

El pin A0 es la entrada a un convertidor analógico digital de 10 bits de resolución, los valores variarán entre 0 y 1024, entonces, en la fórmula es definido como analogRead(PIN_SENSOR), siendo un valor entre 0 < PIN_SENSOR < 1024.

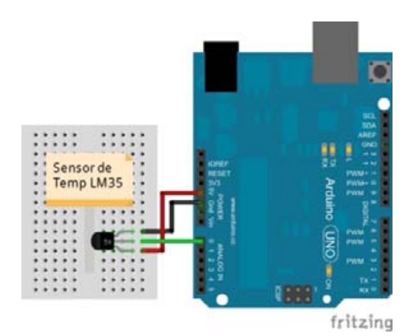


Figura 5. Hardware medición de temperatura

Para obtener el número de revoluciones por minuto a la cual está trabajando el motor, se implementó el Foto interruptor ON1122 (figura 6). La medición de la velocidad se realiza mediante la interrupción de este mismo elemento utilizando un disco perforado (encoder) unido al eje del motor, dicha interrupción se registrara usando la interrupción externa INT1 del ATMEGA ubicada en el pin digital número 3 del Arduino. El valor en RPM se obtiene usando la ecuación (31).

$$vel = \mathrm{i_c Value} * 15$$

Donde la variable $i_c Value$ (32) va a ser igual al número de veces que ha ocurrido una interrupción externa por cada segundo, la magnitud escalar 15 es el resultado de dividir el número de orificios que posee el disco (encoder) ubicado en el motor, entre 60 seg que componen a un minuto con lo que se obtiene la siguiente formula:

$$\frac{60\ segundos}{4\ orificios} = 15$$

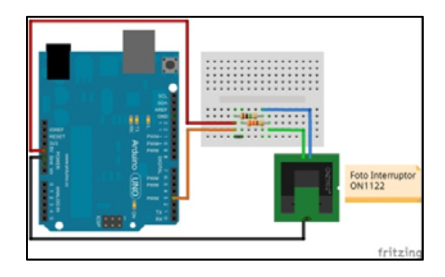


Figura 6. Hardware medición de velocidad.

H. Topología propuesta

Como fue planteado anteriormente en el diseño de control del motor AC, y la topología que debe tener el sistema para poder realizar el control a través de paquetes de datos. En este punto se concatenan todas las propuestas y se define la propuesta final, la cual se explica a continuación:

El control de motor se implementará usando dispositivos electrónicos como los son resistencias, TRIAC NTE56016, Foto-Interruptor ON1122, Opto-acoplador H11AA1, Opto-TRIAC MOC3020, capacitor de 100nf, motor AC monofásico 110V de 500W y un sensor de temperatura LM35 y el Arduino UNO que posee internamente el algoritmo de control del Motor.

La comunicación por paquete de datos es realizada por el Shield SM5100B-D (figura 7), el cual es gobernado por el algoritmo del Arduino UNO que solo contiene el código de comunicación, la supervisión de velocidad y protección Térmica del motor. Este Arduino UNO a su vez comunicara la información procesada y recibida al Arduino UNO que acciona el motor AC, dicha información será transmitida de los pines 0 y 1 (UNO conectado al Shield) a los pines 10 y 11 del Arduino accionador del motor; de igual forma este último Arduino permite observar la ejecución y respuesta del algoritmo casi de manera instantánea al conectar por su puerto USB una PC, la cual debe poseer instalado el software Arduino 1.0.5 o algún visualizador de comunicación como el Hyperterminal. Cabe destacar que para el funcionamiento Shield SM5100B-D se debe conectar una fuente de 5V y que pueda suministrar 2 Amp, ya que a la hora de realizar comunicación el módulo GPRS demanda una corriente pico de 1-2 Amp.

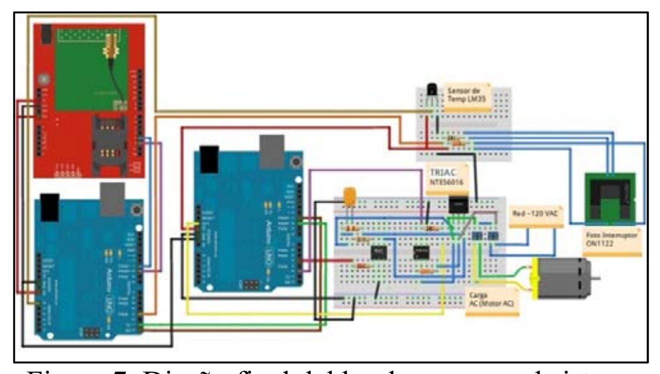


Figura 7. Diseño final del hardware para el sistema

de control

III. RESULTADOS

Como se planteó anteriormente, para realizar las pruebas de conexión del socket se debe configurar inicialmente el APN y la dirección IP del servidor, luego se procede a conectar el Arduino UNO con el puerto USB de una PC para visualizar en ésta la correcta ejecución del código programado, se usó el software HyperTerm de Windows configurado a 9600 baudios, 8 bits de datos, sin paridad, 1 bit de stop y control por software. Y usando los comandos correspondientes se obtienen las respuestas que se muestran a continuación.

Comandos AT q deben ser usados para la correcta configuración:

AT+CGATT=1 (Es para adjuntar servicio de GPRS)

AT+CGDCONT=1,"IP","int.movilnet.com.ve" (Define el APN, en este caso se usa la empresa de telefonía Movilnet)

AT+CGACT=1,1 (Activa el contexto PDP)

AT+SDATACONF=1,"TCP","162.252.57.82",80 (Configura la dirección IP a la cual se realizara conexión TCP)

AT+SDATASTART=1,1 (Inicia la conexión TCP)

AT+SDATASTATUS=1 (Chequea estado del socket)

```
+CGATT: 1

OK
AT+CGDCONT=1,"IP","int.movilnet.com.ve"

OK
Envio de contrasena nula
Activacion de Contexto PDP

OK
Configuracion de la conexion TCP al servidor TCP

OK
Iniciar conexion TCP con Servidor TCP

OK
Chequeando el estado de conexion

+SOCKSTATUS:  1,1,0102,0,0,0

OK
```

Figura 8. Respuesta del SM5100B-D al conectar socket.

```
Configurando contexto PDP
+CGATT: 1
OK
OK
Envio de contrasena nula
Activacion de Contexto PDP
OK
Configuracion de la conexion TCP al servidor TCP
OK
Iniciar conexion TCP con Servidor TCP
OK
Chequeando el estado de conexion
+SOCKSTATUS:  1,0,0104,0,0,0
OK
```

Figura 9. Respuesta del SM5100B-D al no conectar socket.

En las figuras anteriores, se obtuvieron respuestas satisfactorias en relación a la operación del Shield SM5100B-D. En la figura 8 se observa que el módulo SM100B-D se conectó satisfactoriamente al servidor de la página web con la que se trabajó, obteniendo la respuesta (+SOCKSTATUS: 1,0,0102,0,0,0). En la figura 9 se obtiene una respuesta final que el modulo no se conectó satisfactoriamente al servidor arrojando como respuesta (+SOCKSTATUS: 1,0,0104,0,0,0).

Una vez comprobado el correcto funcionamiento del módulo se continuó a verificar que respuesta se obtenida se la página diseñada para iniciar sesión y abrir la página de control con la que se regulara la velocidad del motor AC, con lo que se obtuvo la visual observada en la figura 10. El acceso a esta página se obtiene accediendo a la dirección web (http://salomonjmanosalvag.com.ve).

Figura 10. Página principal.

Al iniciar sesión escribiendo el nombre y contraseña que esta almacenada en la base de datos, se logró direccionar a la página de control del sistema, que presento la visual (figura 11) de la velocidad, temperatura y los botones de control con los que el usuario u operador interactuará.

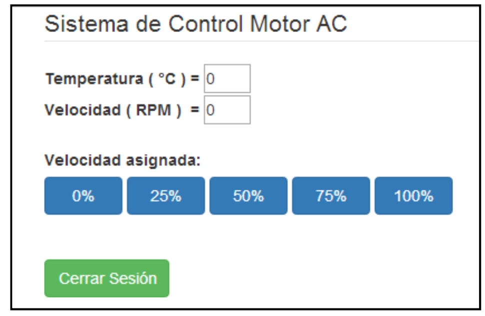


Figura 11. Página de control.

Una vez verificado el correcto funcionamiento de la página web, se procede a enviar información de la velocidad y temperatura del motor usando el módulo SM5100B-D, es importante configurar los distintos parámetros del módulo GPRS antes de transmitir y recibir paquetes de datos, con la excepción de que esta vez al hacer una petición HTTP usando método GET, se deben asignar nuevos parámetros como lo son:

AT+SDATARXMD=1,0,0 Configura el formato en que se recibirá la respuesta del servidor. En esta caso será recibida en HEX.

AT+SDATATSEND=1, String(packetLength) Indica al módulo el tamaño de la data que será enviada, se debe esperar la respuesta del módulo GPRS que define que es momento de introducir el Host, la información a enviar y el user-name usado (Este carácter es ‘>’). Una vez aparezca se inicia el envío de los siguientes datos:

GET/test/insertrow.php?tmp=”+String(temp)+”&vel=”+String(vel)+” HTTP/1.1 (Petición GET que contiene la información)

host = “salomonjmanosalvag.com.ve”
useragent = “Mozilla/5.0”

Luego de introducir estas variables se espera a que aparezca una señalización del módulo inidentificada como ‘+STCPD:1’ que indica que hay respuesta del servidor cono lo que se procede a introducir el comando AT:

AT+SDATAREAD=1 Indica al Módulo GPRS que lea la data recibida en el formato establecido con **AT+SDATARXMD**.

```
Checking socket status:
+SOCKSTATUS:  1,1,0102,0,0,0
OK
Socket connected
Sending HTTP packet...
OK
+SDATA:1,243,485454502F312E3120323030204F4B0D0A446174653A20
+162620266460F726F738D
```

Figura 12. Respuesta al recibir botón 100%.

La respuesta obtenidas mostrada en las figura 12, es el resultado de la respuesta enviadas por el servidor, en respuesta a las peticiones HTTP 1.1 recibidas del módulo GPRS. Estas respuestas del servidor permiten el control de velocidad del motor, con las cuales el Arduino UNO puede ejecutar según el algoritmo contenido en su microcontrolador, la respectiva acción que haya enviado el operador de la página web en ese momento.

Dicho algoritmo identifica el número de caracteres que posee la página web y mediante la función .indexOf, la cual recorre la cadena de datos recibidos del servidor y deduce si se encuentran los caracteres válidos para ejecutar una acción sobre el porcentaje de velocidad al cual debe trabajar el motor AC.

Seguidamente se accede a la página web donde se verifica el que el servidor este recibiendo la información enviada por el Shield SM5100B-D como se observa en la figura 13.



Figura 13. Datos recibidos del SM50100B-D.

Al realizar estas pruebas el tiempo de respuesta después de haber inicializado las configuraciones entre el módulo SM5100B-D y el servidor Web es casi instantáneo solo tardando aproximadamente 1 segundo para recibir la respuesta del servidor, y esta demora es ocasionada por un sleep que está programado en el código del archivo

insertrow.php alojado en el servidor web, ya que sin esta demora el módulo GPRS cerraría casi de manera instantánea la conexión del socket de modo que no se tendría tiempo para leer la data que es enviada por el servidor web.

IV. CONCLUSIONES

Se realizó el estudio de distintas formas de control de un motor AC, resultando el control por fase la más factible ya que ofrece una ventaja en cuanto al tamaño del hardware a implementar, ya que no se tiene la necesidad de diseñar una circuitería complicada para lograr el control del mismo.

La documentación de protocolos de comunicación usando GPRS es obtenida gracias a las facilidades que brinda el manual de usuario que ofrece el fabricante del módulo SM5100B-D.

Al implementar la topología propuesta, la comunicación móvil usando paquete de datos (GPRS) ofreció una buena respuesta a la hora de operar el sistema de control, ya que al enviar la data desde el recinto donde se está controlando el equipo, al servidor externo, la información es recibida casi de manera instantánea. Cabe destacar que gracias al uso de servidores externos de obtienen muchas facilidades a la hora de almacenar data y generar una interfaz visual que sea interactiva con el usuario u operador del sistema.

V. REFERENCIAS BIBLIOGRAFICAS

A. Sano, H. F. (1998). Proceedings IEEE International Conference on Robotics and Automation. *Gain-Scheduled Compensation for Time Delay of Bilateral Teleoperation Systems*, (págs. 1916-1923). Leuven, Belgium.

C. E. García, R. Carelli, J. F. Postigo and B. Morales. (2000). Time Delay Compensation Control Structure for a Robotic Teleoperation System. *4th IFAC International Symposium on Intelligent Components and Instruments for Control.*

Cho, J.-H. P.-C. (1999). Proceedings IEEE/ASME International Conference on Advanced Intelligent Mechatronics. *Sliding-Mode Controller for Bilateral Teleoperation with Varying Time Delay*, (págs. 311-316). Atlanta, USA.

Coiffet, J. V. (1990). *Teleoperation and Robotics. Evolution and Development.* Londres: Kogan Page.

F.-L. Lian, J. Moyne, and D. Tilbury. (2007). Network design consideration for distributed control systems. *IEEE Trans. Contr. Syst. Technol.*, 297-307.

Ferrell, W. R. (1986). Delayed Force Feedback. *IEEE Transactions on Human*, 449 – 455.